
A Physics-preserved Transfer Learning Method for Differential Equations

Hao-Ran Yang

Sun Yat-Sen University
Guangzhou, China

yanghr26@mail2.sysu.edu.cn

Chuan-Xian Ren*

Sun Yat-Sen University
Guangzhou, China

rchuanx@mail.sysu.edu.cn

Abstract

While data-driven methods such as neural operator have achieved great success in solving differential equations (DEs), they suffer from domain shift problems caused by different learning environments (with data bias or equation changes), which can be alleviated by transfer learning (TL). However, existing TL methods adopted in DEs problems lack either generalizability in general DEs problems or physics preservation during training. In this work, we focus on a general transfer learning method that adaptively correct the domain shift and preserve physical relation within the equation. Mathematically, we characterize the data domain as product distribution and the essential problems as distribution bias and operator bias. A Physics-preserved Optimal Tensor Transport (POTT) method that simultaneously admits generalizability to common DEs and physics preservation of specific problem is proposed to adapt the data-driven model to target domain, utilizing the pushforward distribution induced by the POTT map. Extensive experiments in simulation and real-world datasets demonstrate the superior performance, generalizability and physics preservation of the proposed POTT method.

1 Introduction

Many scientific problems, such as climate forecasting [36, 33] and industrial design [40, 2], are modeled by differential equations (DEs). In practice, DEs problems are usually discretized and solved by numerical methods since analytic solutions are hard to obtain for most DEs. However, traditional numerical solvers struggle with expensive computation cost and poor generalization ability. Recently, dealing DEs with deep neural network has attracted extensive attention. These methods can be roughly divided into two categories: physics-driven and data-driven. Optimizing neural networks with objective constructed by exact equations, physics-driven methods such as Physics-Informed Neural Networks [28, 26] have great interpretability, but suffer from the poor generalization capability across equations and the hard requirement of exact formulation of DEs. In contrast, Data-driven methods such as neural operator [24, 22] typically take the alterable function in the equations as input data and solution function as output data. Their generalization capability are markedly improved as they can cope with a family of equations rather than one.

However, the performance of data-driven methods are highly dependent on identical assumption of training and testing environments. If the testing data comes from different distribution, model performance may degrade significantly. In practice, however, applying model to different data distributions is a common requirement, e.g. from simulation data to experiment data, cross-region model application. While it is often hard to collect sufficient data from a new data domain to train a new model, transfer learning (TL) that aims to transfer model from source domain with plenty of data

*Corresponding author.

Table 1: Simulation datasets used for experiments. Figures in $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ are examples of (input,output) pairs from different domains in the transfer tasks. For 1-d curve plot, the filled regions represent the areas between the curve and the x-coordinate. For 2-d surface plot, the pixel value at each image pixel corresponds to the function value at the sampling point. Brighter color indicates larger value. Definition and more details about the subdomains datasets are provided in Appendix C.1.

EQUATIONS	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$
BURGERS' EQUATION $u_t + uu_x = \nu u_{xx}$, $\mathcal{G}_\theta : u(x, 0) \rightarrow u(x, 1)$			
ADVECTION EQUATION $u_t + \nu u_x = 0$, $\mathcal{G}_\theta : u(x, 0) \rightarrow u(x, t)$			
DARCY FLOW $\nabla \cdot (k(x)\nabla u(x)) = 1$, $\mathcal{G}_\theta : k(x) \rightarrow u(x)$			

to target domain with inadequate data, is widely adopted in real-world applications [39]. However, TL methods for DEs problems are still unexplored.

In this work, we carefully analyze the transfer learning settings in DEs problems and modelize the essential problem as distribution bias and operator bias. Given such perspective, we fully investigate the existing TL methods used in DEs problems. Technically, they can be summarized as three types. **(1) Analytic methods** [9] induce an analytic expression about model parameters to achieve adaptation with a few samples. Nevertheless, they are only feasible in very few problems with nice properties and hence not a general methodology. **(2) Finetuning** [31] the well-trained source model by target data. It is widely-used in DEs problems with domain shift due to its simplicity. Nevertheless, when the amount of available target data is limited, directly correcting operator bias by finetuning is insufficient. **(3) Domain Adaptation (DA)** [34] methods developed in other areas such as computer vision. They typically align feature distributions of source and target domains and remove the domain-specific information, so the aligned feature together with the model trained from them are domain-invariant. However, the physical relation of DEs may not necessarily be valid in the aligned feature space. Among these methods, analytic methods and finetuning directly correct the operator bias while the DA methods focus on the distribution bias correction. But all these methods have certain limitations. Therefore, a general model transfer method that can preserve the physical relation is worth exploring.

Our idea is to characterize the target domain with physics preservation and then fully correct the operator bias. Briefly speaking, we propose the Physics-preserved Optimal Tensor Transport (POTT) method to learn a physics-preserved optimal transport map between source and target domains. Then the target domain with the physical relations are characterized by the pushforward distribution, which enables a more comprehensive training for model transfer learning. Thus, the model's generalization performance on target domain can be largely improved even when only a small number of target samples are available for training. To encourage the POTT map to characterize target distribution in a physics-preserved way, we introduce a problem-specific physical regularization to the OT problem, which is derived from available physical prior of the problem. In general cases without physical prior, the regularization term is formulated as relation between marginal pushforward distribution. Our contributions are summarized as follows:

- A detailed analysis of transfer learning for DEs problems is presented, based on which we propose a feasible transfer learning paradigm that simultaneously admits generalizability to general DEs problems and physics preservation of specific problems.
- We propose POTT method to adapt the data-driven model to target domain with the pushforward distribution induced by the POTT map. A dual optimization problem is formulated to explicitly solve the optimal map. The consistency property between the solution and the ideal optimal map is presented.
- Extensive evaluation and analysis experiments on both simulation and real-world datasets are conducted. POTT shows superior performance on different types of equations with transfer tasks

of varying difficulties. Intuitive visualization analysis further supports our discussion on the physics preservation of POTT.

2 Preliminary

Data-driven methods for DEs problems. In DEs problems, data-driven methods aim to learn functional maps from data distributions. DeepONet [24] is proposed based on the universal approximation theorem [4]. Then the MIONet [16] further extends DeepONet to problems with multiple input functions and Geom-DeepONet [15] enables DeepONet to deal with parameterized 3D geometries. Differently, FNO [22] is constructed in the insight of approximating integration in the Fourier domain. Geo-FNO [21] extends FNO to arbitrary geometries by domain deformations. F-FNO [32] enhances FNO by employing factorization in the Fourier domain. Recently, transformer have also been used to construct neural operators [19, 14, 20, 35]. Although having achieved great success, these data-driven methods induce a common issue: they are highly dependent on identical assumption of training and testing environments. If the testing distribution differs, the performance of the neural operators will significantly degrade.

Transfer learning. Most of the transfer learning methods are proposed for Unsupervised Domain Adaptation (UDA) with classification task. They align the feature distributions of source and target domain by distribution discrepancy measurement [23], domain adversarial learning [11, 3], etc. Recent methods [5, 27, 38] further extend DA to regression settings with continuous variables. For DEs problems, analytic transfer methods [9] are presents for specific equations; finetuning [37, 31] and DA methods [12, 34] are applied in various tasks. However, there isn't a general physics-preserved transfer learning method developed for DEs problems.

Optimal transport (OT). OT has been quite popular in machine learning area [7, 6]. The most widely known OT problems are the Monge problem and the Kantorovich problem defined as follows:

$$M(P^s, P^t) = \inf_{T_{\#}P^s=P^t} \int_{\Omega^s} c(x, T(x)) dP^s(x) \quad (1)$$

$$K(P^s, P^t) = \inf_{\pi \in \Pi(P^s, P^t)} \int_{\Omega^s \times \Omega^t} c(x, y) d\pi(x, y), \quad (2)$$

where $c(x, y)$ denotes the cost of transporting $x \in \Omega^s$ to $y \in \Omega^t$, $T : \Omega^s \rightarrow \Omega^t$ denotes the transport map, $T_{\#}P^s$ denotes the pushforward distribution, and $\Pi(P^s, P^t)$ denotes the set of joint distributions with marginal P^s and P^t . The Monge problem aims at a transport map $\mathcal{T}$ that minimize the total transport cost, called the Monge map. However, usually the solution of the Monge problem does not exist, so the relaxed Kantorovich problem is more widely used. The Kantorovich problem can be solved as a linear programming problem. With an entropic regularization added, it can be fastly computed via the Sinkhorn algorithm [7]. Moreover, if π^* takes the form $\pi^* = [id, \mathcal{T}]_{\#}P^s \in \Pi(P^s, P^t)$, then $\mathcal{T}$ is the Monge map. However, these optimization method don't scale well to large scale data domain and can't handle continuous probability distributions [30]. To deal with these limitations, neural OT methods are developed [30, 8, 18]. They typically train the neural network to directly approximate the OT map via constructing objective function by various OT problems [10, 13, 1].

3 Analysis and Motivation

3.1 Problem Formulation and Notations

Now we formally formulate the transfer learning settings for DEs problems. Consider two function space $\mathcal{D}_k, \mathcal{D}_u$ with elements $k : \Omega_k \rightarrow \mathbb{R}, u : \Omega_u \rightarrow \mathbb{R}$. Denote the product spaces as $\mathcal{D} = \mathcal{D}_k \times \mathcal{D}_u$ and $\Omega = \Omega_k \times \Omega_u$. Suppose there exist physical relations within the product space $\mathcal{D}$, which can be characterized as the following two forms:

$$\mathcal{F}(k, u) = 0 \text{ (Equation form)} \quad (3)$$

$$\mathcal{G}(k) = u \text{ (Operator form)} \quad (4)$$

Obviously, relation Eq. (4) is the explicit form of the implicit operator mapping determined by Eq. (3), whose existence is theoretically guaranteed under some conditions such as the implicit function theorem. Once the operator $\mathcal{G} : \mathcal{D}_k \rightarrow \mathcal{D}_u$ is solved, we can predict the desired physics quantities

u for a group of k . However, directly solving the implicit function from Eq. (3) is often extremely difficult. In such cases, fitting $\mathcal{G}$ by neural network with collected data set $\{(k, u)\}$ provides a practical way for numerical approximation, which is exactly the goal of data-driven methods. Here we slightly abuse the notations k and u to represent both functions and their discretized value vectors.

An essential limitation is that the learning of the operator network $\hat{\mathcal{G}}$ depends heavily on the distribution of collected data. Let $P^s, P^t \in \mathcal{P}_{\mathcal{D}}$ be two product distributions supported on the source and target domain $\mathcal{D}^s, \mathcal{D}^t \subset \mathcal{D}$, respectively. Then the operator trained from them, denoted as $\hat{\mathcal{G}}^s$ and $\hat{\mathcal{G}}^t$, are in fact the approximations of $\mathcal{G}^s := \mathcal{G}|_{\mathcal{D}^s}$ and $\mathcal{G}^t := \mathcal{G}|_{\mathcal{D}^t}$. When distribution shift occurs, the operator relation also differs. So the transfer learning problem for DEs can be modeled as

$$\begin{aligned} P^s(k, u) &\neq P^t(k, u), & (\text{Distribution bias}) \\ \implies \mathcal{G}^s &\neq \mathcal{G}^t. & (\text{Operator bias}) \end{aligned} \quad (5)$$

In these situations, the model performance generally degrades if $\hat{\mathcal{G}}^s$ is directly applied to $\mathcal{D}^t$. While collecting sufficient training data is difficult in many applications scenarios, a common requirement is to transfer $\mathcal{G}^s$ to $\mathcal{D}^t$ with a few target data available. Formally, given $\hat{\mathcal{D}}^s = \{(k_i^s, u_i^s)\}_{i=1}^{n^s}$, $\hat{\mathcal{D}}^t = \{(k_j^t, u_j^t)\}_{j=1}^{n^t}$, with $n^t \ll n^s$, the task is to transfer source model $\hat{\mathcal{G}}^s$ to target domain $\mathcal{D}^t$ and approximate $\mathcal{G}^t$, i.e. to correct the operator bias. An intuitive illustration is shown in Fig. 2.

3.2 Methodology Analysis

Based on problem 5, existing transfer learning methods either directly correct the operator bias or indirectly correct the operator bias by aligning the feature distributions, all of which are subject to certain limitations.

Analytic methods directly correct the operator bias by deriving analytic expressions

$$\hat{\mathcal{G}}^t = \mathcal{H}_a(\hat{\mathcal{G}}^s, \hat{\mathcal{D}}^t), \quad (6)$$

where $\mathcal{H}_a$ denotes the ideal analytic formulation. Although exhibiting excellent interpretability, they are limited to problems with nice property and sufficient priors such as the explicit form of the equation. So they can only be applied to few problems with well-behaved equations.

Finetuning by target domain data directly corrects the operator bias by only further training the source model with collected target data:

$$\hat{\mathcal{G}}^t = \min_{\mathcal{G}} \mathcal{L}_{\text{task}}(\hat{\mathcal{D}}^t; \hat{\mathcal{G}}^s), \quad (7)$$

where $\mathcal{L}_{\text{task}}$ denotes the task-specific training loss. It does not actively and fully leverage the knowledge of source and target domain data. When the amount of available target data is limited, the predictions of target samples exhibit characteristics similar to the source samples since inadequate data is insufficient for model transfer, as discussed in Sec. 5.

Distribution alignment methods from DA indirectly correct the operator bias by aligning the feature distributions. These methods typically aim to learn a feature map and a corresponding feature space in which the distance between source and target feature distributions is minimized. Then a predictor trained by the source domain features can be expected to perform well on target domain features:

$$g^* = \min_g \text{dist}(g_{\#}P^s, g_{\#}P^t), \quad \hat{\mathcal{G}}^t = \min_{\mathcal{G}} \mathcal{L}_{\text{task}}(g^*(\hat{\mathcal{D}}^s) \cup g^*(\hat{\mathcal{D}}^t); \hat{\mathcal{G}}^s), \quad (8)$$

where $\text{dist}(\cdot, \cdot)$ denotes a measurement of distribution discrepancy, g is the learned feature map, $g_{\#}P^s, g_{\#}P^t$ are the pushforward feature distributions. DA method is purely data-driven without the need of physical priors so it is a general methodology that can be used in most scenarios. However,

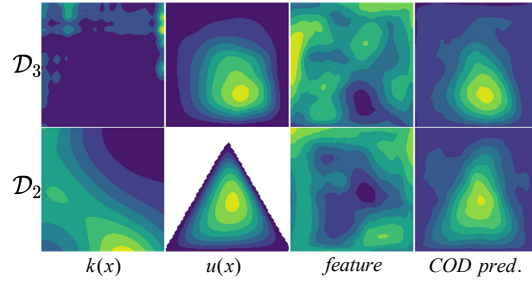


Figure 1: Visualization of DA method (COD) in task $\mathcal{D}_3 \rightarrow \mathcal{D}_2$ on Darcy flow. The 1st and 2nd columns present the input $k(x)$ and output $u(x)$ sample pairs of $\mathcal{D}_3$ and $\mathcal{D}_2$. The 3rd column visualizes their feature maps from the aligned distributions, which are forced to be analogous but lacks clear structure explicit physical meaning. It is unclear whether they retain the correct physical information. The prediction shown in the 4th column verifies that the physical structures of u are not fully preserved.

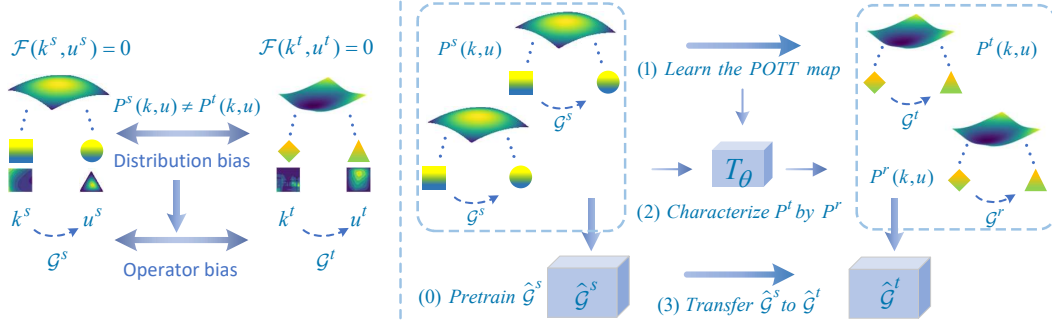


Figure 2: Illustration of POTT. **Left:** Illustration of problem formulation. The distribution bias leads to the bias of physics, i.e. the operator bias. The goal is to correct the operator bias. **Right:** POTT correct the operator bias by characterizing $\mathcal{D}^t$ in a physics-preserved way. (0) Before the model transfer process, source model $\hat{\mathcal{G}}^s$ is pretrained with sufficient source data. (1) The POTT map T_θ between source and target domain is learned. (2) The target distribution P^t is characterized by the pushforward distribution P^r . (3) $\hat{\mathcal{G}}^s$ is transferred to $\hat{\mathcal{G}}^t$ with $\hat{\mathcal{D}}^t$ and $\hat{\mathcal{D}}^r$.

the two feature distributions are aligned by removing the domain specific knowledge so the domain invariant representations are obtained. In other words, the aligned feature distribution may lose some domain specific physical relations of both domain. As shown in Fig. 1, the features of source and target samples are analogous but confused. It is unclear whether the physical relations are preserved, which is also indicated by the output of learned $\hat{\mathcal{G}}^t$.

Motivation of POTT. Generally, directly correcting operator bias by finetuning only partially transfers with limited target data, while indirectly operator bias via feature distribution alignment may distort the physical relations of the DEs problem. Therefore, we propose to correct the operator bias by characterizing the target domain with physics preservation and then fully transferring $\hat{\mathcal{G}}^t$ to $\mathcal{D}^t$:

$$\hat{\mathcal{D}}^r = \mathcal{H}_c(\hat{\mathcal{D}}^s, \hat{\mathcal{D}}^t, \mathcal{R}), \quad \hat{\mathcal{G}}^t = \min_{\mathcal{G}} \mathcal{L}_{\text{task}}(\hat{\mathcal{D}}^r \cup \hat{\mathcal{D}}^t; \hat{\mathcal{G}}^s), \quad (9)$$

where $\mathcal{H}_c$ characterizes $\mathcal{D}^t$ by collected dataset $\hat{\mathcal{D}}^s$, $\hat{\mathcal{D}}^t$ and the physical regularization $\mathcal{R}$.

4 POTT Method

The major obstacle in Eq. (9) is to construct the $\mathcal{H}_c$. A practical way is to learn a transformation $\mathcal{T}$ between P^s and P^t , so the target distribution P^t can be characterized by the pushforward distribution $P^r = \mathcal{T}_\# P^s$, i.e. $\mathcal{H}_c(\cdot) = \mathcal{T}(\hat{\mathcal{D}}^s; \hat{\mathcal{D}}^t, \mathcal{R})$. Note that the exact corresponding relations between samples from P^s and P^t is unknown, so it is impractical to train $\mathcal{T}$ by traditional supervised learning. In other words, $\mathcal{T}$ shall be trained via an unpaired sample transformation paradigm. In OT theory, an optimal transport map between two distributions is the solution of an OT problem and the computation of the OT problem does not need paired samples. Therefore, a natural idea is to **model the ideal map $\mathcal{T}$ by an OT map between P^s and P^t** . In the following sections we regard the desired map as the OT map $\mathcal{T}$, distinguishing from map T that is not necessary optimal.

4.1 Formulation of POTT

A brief review of OT problem is provided in Sec. 2. As our purpose is to characterize P^t by $P^r = \mathcal{T}_\# P^s$, we focus on the Monge problem Eq. (1). Moreover, the physical relation in DEs can be regarded as relation between P_k and P_u . Thus, a more reasonable perspective is to consider an OT problem between two product distributions, known as the Optimal Tensor Transport (OTT) problem. In this perspective, we propose the physics-preserved optimal tensor transport (POTT) problem:

Definition 4.1 (POTT). Given two product distributions $P^s, P^t \in \mathcal{P}_{\mathcal{D}_k \times \mathcal{D}_u}$, define the physics-preserved optimal tensor transport (POTT) problem as:

$$\inf_{T_\# P^s = P^t} \int_{\mathcal{D}_s} c((k, u), T(k, u)) dP^s + \mathcal{R}(T), \quad (10)$$

where $T = (T_k, T_u) = (T|_{\mathcal{D}_k}, T|_{\mathcal{D}_u})$, $\mathcal{R}(T) = \mathcal{R}(T_k, T_u)$ is the physical regularization.

Specifically, when the pushforward distribution P^r perfectly matches the target distribution P^t , the physical relation within is automatically obtained. However, it is hard to achieve in practice and P^r should be regarded as an approximation or a disturbance of P^t . Then we expect P^r to approach P^t in a physics-preserved way at least, which relies on the physical regularization $\mathcal{R}$. The construction of $\mathcal{R}$ depends on the specific problem. Here we provide two basic ideas for example:

- If some physical priors of the problem are available, $\mathcal{R}$ can be derived from the priors. For example, when the weather forecasting problem is modeled as an advection partial differential equation [33], the value preservation property $\int u(x, t)dx = \text{const}, \forall t$, is a strong inductive bias. Then the $\mathcal{R}$ can be formulated as the variance of the system's value:

$$\mathcal{R}(T) = \text{var} \left(\int T_u u^s(x, t)dx \right), \quad (11)$$

Experiment of this problem is shown in Sec. 5.

- For the general cases with no physical priors available, we formulate $\mathcal{R}$ as the physical relation between the marginal distributions of P^r :

$$\mathcal{R}(T) = \mathcal{R}(T_k, T_u) = m(\mathcal{G}(k^r), u^r) = m(\mathcal{G}T_k(k^s), T_u(u^s)), \quad (12)$$

where the $m(\cdot, \cdot)$ can be any metric on $\mathcal{D}_u$. An alternative is the L_2 norm in the vector space. In practice, the operator $\mathcal{G}$ can be substituted by $\hat{\mathcal{G}}^s$ or $\hat{\mathcal{G}}^t$ as an approximation.

Remark 4.2. While both PINNs-based methods and POTT emphasize the integration of physics into the learning process, they are not directly comparable. As described in Sec. 1, PINNs-based methods rely on the explicit form of the equations to define the physics-informed loss. Consequently, they become inapplicable when the underlying equations are unknown or only partially specified. In contrast, by leveraging the optimal transport framework and incorporating physical regularization, POTT can be considered as a combination of data-driven and physics-driven method and is applicable to a broader range of problems.

4.2 Optimization and Analysis

Existing OTT methods [17] mainly focus on the discrete case of entropy-regularized OTT and solve the optimization problem via the Sinkhorn algorithm, which is not suitable for continuous OTT with physical regularization. Motivated by neural OT methods [30, 18], we explicitly fit $\mathcal{T}$ by a neural network and optimize it with the gradient of training loss. But Eq. (10) is a constrained optimization problem and it is challenging to satisfy the constraint during the optimization process. Therefore, we introduce the Lagrange multiplier and reformulate Eq. (10) to the unconstrained dual form.

$$\sup_f \inf_T \int_{\Omega^s} c((k, u), T(k, u)) - f(T(k, u)) + \lambda \mathcal{R}(T) dP^s + \int_{\Omega^t} f(k, u) dP^t. \quad (13)$$

Optimization with gradient descent tends to converge to a saddle point (T^*, f^*) . Following previous work [10], the consistency between T^* and $\mathcal{T}$ is guaranteed.

Theorem 4.3 (Consistency). *Suppose the dual problem Eq. (13) admits at least one saddle point solution, denoted as (T^*, f^*) . Let $\mathcal{L}$ be the objective of Eq. (13). Then*

- the dual problem Eq. (13) equals to the Kantorovich problem with physical regularization in terms of total cost, i.e. $\mathcal{L}(P^s, T^*, f^*) = K(P^s, P^t) + \mathcal{R}(T^*)$.
- if $T^*_{\#} P^s = P^t$, then Eq. (13) degenerates to the dual form of the primal Monge problem Eq. (1), T^* is a Monge map, i.e. $\mathcal{L}(P^s, T^*, f^*) = M(P^s, P^t)$.

The proof of Thm 4.3 can be found in Appendix B. Theoretically, if the Monge map exists, i.e. $P^r = P^t$, then P^r automatically admits the physics contained in P^t . The solution of the Monge problem is the desired OT map. However, as mentioned in Sec. 4.1, in most cases the Monge map does not exist. Therefore, the saddle point T^* is not an optimal solution of the physics-regularized Monge problem. In this situation, Thm 4.3 states that Eq. (13) equals to physics-regularized Kantorovich problem in terms of total cost. Thus, the learned T^* can be regarded as a compromise solution between the Monge problem and the Kantorovich problem. Importantly, the physical regularization encourages physics preservation during the training process of T , which is crucial for the approximation of P^t .

Table 2: Evaluation results of Burgers' equations.

METHOD	$\mathcal{D}_1 \rightarrow \mathcal{D}_2$		$\mathcal{D}_1 \rightarrow \mathcal{D}_3$		$\mathcal{D}_3 \rightarrow \mathcal{D}_2$		AVERAGE	
	50	100	50	100	50	100	50	100
TARGET ONLY	0.2142	0.1429	0.1173	0.0968	0.2142	0.1429	0.1819	0.1275
SRC+TGT	0.3960	0.3982	0.3486	0.3350	0.3145	0.2930	0.3530	0.3421
FINETUNING	0.2001	0.1191	0.1049	0.0801	0.1546	0.0938	0.1532	0.0977
TL-DEEPONET	0.1623	0.1182	0.1275	0.1127	0.1763	0.1436	0.1554	0.1248
DARE-GRAM	0.1727	0.1145	0.1241	0.1099	0.1752	0.1393	0.1573	0.1212
COD	0.1713	0.1225	0.1288	0.1105	0.1818	0.1525	0.1606	0.1285
POTT	0.1528 ± 0.016	0.0965 ± 0.012	0.0950 ± 0.007	0.0705 ± 0.008	0.1249 ± 0.015	0.0757 ± 0.019	0.1242 ± 0.013	0.0809 ± 0.013

Table 3: Evaluation results of Darcy flow.

TASK METHOD	$\mathcal{D}_2 \rightarrow \mathcal{D}_1$		$\mathcal{D}_1 \rightarrow \mathcal{D}_3$		$\mathcal{D}_2 \rightarrow \mathcal{D}_3$		AVERAGE	
	50	100	50	100	50	100	50	100
TARGET ONLY	0.1815	0.1122	0.3925	0.2893	0.3925	0.2893	0.3222	0.2303
SRC+TGT	0.7113	0.7600	0.1581	0.1409	0.3535	0.2381	0.4076	0.3797
FINETUNING	0.1426	0.0869	0.1556	0.1605	0.4693	0.3553	0.2558	0.2009
TL-DEEPONET	0.1410	0.0805	0.1539	0.1481	0.4514	0.2842	0.2488	0.1709
DARE-GRAM	0.1395	0.0805	0.1533	0.1441	0.4509	0.2842	0.2479	0.1696
COD	0.1367	0.0794	0.1527	0.1481	0.4437	0.2836	0.2444	0.1704
POTT	0.1362 ± 0.002	0.0762 ± 0.002	0.1397 ± 0.009	0.1404 ± 0.006	0.3527 ± 0.025	0.2271 ± 0.019	0.2095 ± 0.012	0.1479 ± 0.009

In practice, we parametrize the map T , dual multiplier f and the operator $\mathcal{G}$ by neural networks T_θ , f_ϕ and $\mathcal{G}_\eta$ with parameters denoted by θ , ϕ and η . Function variables k and u are discretized into vectors. The overall objective of POTT method is

$$\begin{aligned}
& \max_{\phi} \min_{\theta} \sum_{i=1}^{n^s} c((k_i^s, u_i^s), T_\theta(k_i^s, u_i^s)) - f_\phi(T_\theta(k_i^s, u_i^s)) + \lambda \mathcal{R}(T_\theta) + \sum_{j=1}^{n^t} f_\phi(k_j^t, u_j^t) \\
& \min_{\eta} \sum_{j=1}^{n^t} \hat{\mathcal{L}}_{\text{task}}(\hat{\mathcal{G}}_\eta^t(k_j^t, u_j^t) + \beta \sum_{i=1}^{n^s} \hat{\mathcal{L}}_{\text{task}}(\hat{\mathcal{G}}_\eta^t(k_i^r, u_i^r)),
\end{aligned} \tag{14}$$

where $k_i^r = T_{\theta_k}(k_i^s)$, $u_i^r = T_{\theta_u}(u_i^s)$. $\hat{\mathcal{L}}_{\text{task}}$ is the task specific loss. λ and β are hyper-parameters. Physical regularization term $\mathcal{R}$ depends on the available physical priors, as discussed in Sec. 4.1. A form of algorithm and the discussion of the computational cost are provided in Appendix C.

5 Experiment

Benchmarks. Experiments are conducted on both simulation and real-world datasets. For simulation datasets, three representative equations are contained. For real-world datasets, we consider the cross-region climate forecasting task. Implementation details and definition of RMSE are provided in Appendix C.

- **Simulation dataset.** Following previous works [22, 25], we take the 1-d Burgers' equation, 1-d space-time Advection equation, and 2-d Darcy Flow problem as our benchmarks. A brief introduction of these DEs problems can be found in Tab. 1. To simulate the domain shift, three different sub-domains for each equation, denoted as $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, are generated. We generate 1000 training samples for each domain of Burgers' equation and 2000 samples for Advection equation and Darcy Flow. To fully investigate the effectiveness of transfer learning methods, we consider two scenarios that only 50 and 100 target data samples are available for model transfer.

Table 4: Evaluation results of Advection equations.

METHOD	$\mathcal{D}_1 \rightarrow \mathcal{D}_2$		$\mathcal{D}_2 \rightarrow \mathcal{D}_1$		$\mathcal{D}_3 \rightarrow \mathcal{D}_2$		AVERAGE	
	50	100	50	100	50	100	50	100
TARGET ONLY	0.2162	0.1261	0.2585	0.1382	0.2162	0.1261	0.2303	0.1301
SRC+TGT	0.2347	0.1400	0.7299	0.4041	0.3969	0.1574	0.4538	0.2338
FINETUNING	0.0247	0.0143	0.2193	0.0891	0.1257	0.0723	0.1532	0.0977
TL-DEEPONET	0.0587	0.0127	0.2365	0.1047	0.1534	0.0685	0.1495	0.0620
DARE-GRAM	0.0572	0.0121	0.2227	0.0805	0.1687	0.0700	0.1495	0.0542
COD	0.0530	0.0120	0.2252	0.0785	0.1593	0.0644	0.1458	0.0516
POTT	0.0207	0.0112	0.1872	0.0787	0.1016	0.0613	0.1032	0.0504
	± 0.004	± 0.003	± 0.026	± 0.017	± 0.015	± 0.007	± 0.015	± 0.009

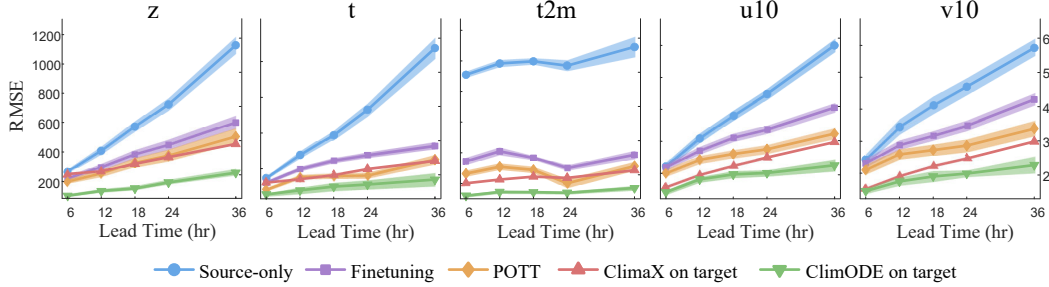


Figure 3: Comparison with (1) Source-only, source pretrained model, (2) Finetuning on target domain, (3) Climax, a SOTA climate forecasting model, trained with sufficient target data, (4) ClimODE, the backbone model, trained with sufficient target data. The light area reflects the standard deviation of RMSE ($\downarrow$). Details are provided in Tab. 14.

For all transfer tasks, we use 10 extra target domain samples for validation and 100 for testing. Note that the necessity of transfer learning depends on the extent of domain shift. For the transfer tasks with minor domain shift, source model can generalize well and additional transfer learning methods are unnecessary. Therefore, we only evaluate these methods in some hard tasks that need more proactive transfer. Details about data generation and how the domain shift between sub-domains is considered are provided in Tab. 8, 10, 12, and Tab. 9, 11, 13 in Appendix C.1. Note that **no equation-specific information (e.g., symbolic expressions or coefficients) is provided and only (k, u) data pairs are available during training**, which mimics real-world conditions where governing equations are unknown. Therefore, the physical regularization $\mathcal{R}$ in Eq. (12) is used for all these problems.

- **Real-world dataset.** To better assess the potential of POTT in cross-domain application of data-driven methods, we evaluate POTT in the cross-region climate forecasting task. The preprocessed 5.625° resolution ERA5 dataset from WeatherBench [29] is used for evaluation and the SOTA ClimODE [33] is used as backbone model. Following [33], we consider 5 quantities as label variables: ground temperature (t2m), atmospheric temperature (t), geopotential (z), and ground wind vector (u10, v10). We use climate data on North America as source domain and the global climate data as target domain. For source domain, we use ten years of training data (2006-15). For target domain, only one year (2015) of training data is available, while data of 2016 is used for validation and data of 2017-18 is used as testing data. More details are provided in Appendix C. As the ClimODE is used as backbone model, we follow the idea that modeled the climate system as an advection equation and use the physical regularization in Eq. (11) for POTT.

Comparison methods As discussed in Sec. 3.2, analytic methods are hard to apply in general data-driven DEs methods. So we compare POTT with finetuning and DA methods, including **Finetuning** on source pretrained model; **Src+Tgt**, training from scratch with source and target data; **TL-DeepONet** [12], a representative DA method proposed for DEs problems; **DARE-GRAM** [27] and **COD** [38], two SOTA DA methods proposed for DA regression problem with continuous variables. Since DARE-GRAM and COD are unsupervised DAR methods proposed for tasks in computer vision, we add the supervised target loss to them for fairness.

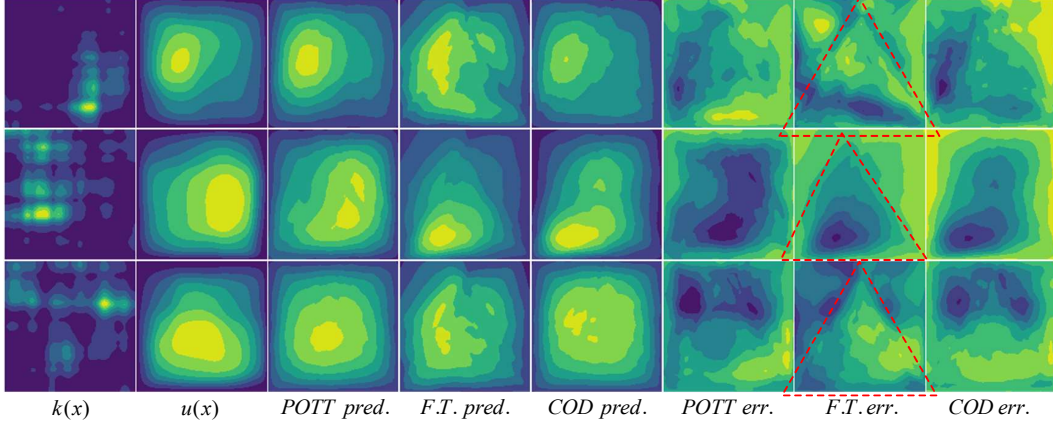


Figure 4: Visualization of $\hat{G}_t$ on Darcy Flow. The value at each image pixel represents the function value at the sampling point. Brighter colors (yellow) indicate higher values. Columns 1-2 show the input-output function pairs from the target domain. Columns 3-5 show the output of POTT, finetuning (F.T.) and COD. Columns 6-8 display the prediction errors relative to the ground truth $u(x)$.

Evaluation results. For simulation datasets, average rMSE of three times repeated experiments are reported. For climate forecasting, the mean and standard deviation of latitude-weighted RMSE is reported. Details about the evaluation metrics are shown in Appendix C.

- **Simulation datasets.** As shown in Tab. 3, POTT outperform finetuning and DA methods in most tasks. In tasks with severe domain shift such as $\mathcal{D}_2 \rightarrow \mathcal{D}_3$ of Darcy flow, the performances of existing methods are not satisfactory, while POTT reduces the relative error by nearly 25% (from 0.4693 to 0.3527) with only 50 target samples available, and reduces the relative error by 36.08% (from 0.3553 to 0.2271) with 100 target samples. In simpler tasks $\mathcal{D}_2 \rightarrow \mathcal{D}_1$ and $\mathcal{D}_1 \rightarrow \mathcal{D}_3$, POTT still reduce the relative error compared to finetuning and DAR methods in every task. Results of Burgers' equation and Advection equation are provided in Tab. 2 and Tab. 4.
- **Climate forecasting.** We only compare POTT with finetuning method because the DA methods are not easily applicable for the model architecture of ClimODE. As shown in Fig. 3, directly applying ClimODE trained on North America to global forecasting results in distinct prediction error. The model performances are improved by finetuning, and POTT further reduce the prediction error, approaching the ClimODE model trained with sufficient target data. Note that the prediction error of POTT is quite closed to Climax model trained with sufficient target data, which is also a SOTA model in climate forecasting. Such experiment demonstrates the potential of POTT as a general model transfer method for DEs problems and real-world applications.

Visualization analysis of prediction. Fig. 4 illustrates the outputs and error maps for the target sample predicted by POTT, finetuning, and COD on the Darcy $\mathcal{D}_3 \rightarrow \mathcal{D}_2$ task with 100 target samples. (1) As shown in columns 3 – 5, despite only 100 target samples are available for training, the shape and the variation trend of outputs predicted by POTT are consistent with the ground truth. In contrast, predictions of finetuning indicate that the transferred model fails to learn the right shape and variation trend. Predictions of COD are globally consistent with ground truth, but the transferred model fails to correctly predict the large value areas. (2) In the error maps shown in columns 6 – 8, the bright areas in the error maps of POTT are the smallest among the three methods. Especially, the error maps of finetuning shown in 7th column clearly exhibit the characteristics of source domain distribution, i.e., the distinct triangular patterns, supporting the discussions in Sec. 3.2.

Ablation analysis of physical regularization. To investigate the effect of the physical regularization in Eq. (10), we implement an ablation analysis on the task $\mathcal{D}_1 \rightarrow \mathcal{D}_2$ of Darcy flow. As shown in Fig. 5, the outputs of OTT roughly resemble the ground truth u_{tgt} in the regions of large values, but the triangular structure is not preserved, indicating the loss of certain physical

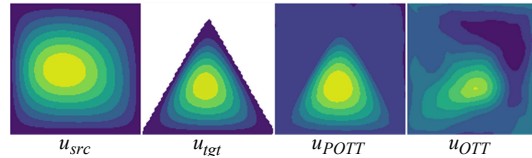


Figure 5: Visualization of POTT and OTT.

relations. In contrast, with physical regularization, the outputs of POTT exhibit consistency in both the regions of large values and the triangular structure, verifying the preservation of physics.

To more intuitively demonstrate the effect of physical regularization, we present the numerical results of $\hat{\mathcal{G}}_t$ trained using the outputs of OTT and POTT, as shown in Tab. 5. It is observed that POTT achieves lower prediction error than OTT, indicating that the outputs of POTT better characterize the target distribution in a physics-preserved manner. Furthermore, both OTT and POTT outperform vanilla finetuning.

$\mathcal{D}_1 \rightarrow \mathcal{D}_2$	Finetuning	POTT w/o $\mathcal{R}$	POTT
	0.0575	0.0543	0.0496

Table 5: Results of the ablation analysis.

Extreme few-shot settings.

In extreme few-shot settings (e.g., less than 10 target samples), both finetuning and POTT indeed face significant challenges due to the risk of overfitting. From a

n^t	100	50	20	10	5	1
Tgt. only	0.3943	0.3925	0.4130	0.4897	0.5107	0.5568
Finetuning	0.3553	0.4693	0.4866	0.5368	0.5521	0.6282
POTT	0.2271	0.3527	0.4126	0.5257	0.5364	0.6359

Table 6: Results of scale-down experiment.

distribution-matching perspective, fitting a meaningful target distribution with only a few samples is fundamentally ill-posed and prone to instability. This limitation applies broadly to all distribution-based methods. To empirically evaluate this, we conduct a scale-down experiment on the Darcy flow task $\mathcal{D}_2 \rightarrow \mathcal{D}_3$ under various numbers of labeled target samples n^t . As shown in Tab. 6,

- As n^t decreases, all methods degrade in performance.
- When $n^t \geq 20$, POTT still provides advantages over both finetuning and training from scratch.
- When $n^t \geq 10$, both POTT and finetuning begin to overfit, and performance starts to deteriorate significantly.
- At the extreme case of $n^t = 1$, POTT fails entirely, as the transport map becomes unreliable and may even hurt the learning of the target predictor due to inaccurate distribution matching.

These results clarify that POTT is most effective in the moderate few-shot regime, but may not be applicable when labeled target data is extremely scarce, which aligns with many real-world cases in scientific machine learning where acquiring a small number of high-quality labels is feasible.

6 Conclusion

In this work, we studied the domain shift issue in DEs problems. The essential problem is modeled as distribution bias and operator bias. Then we detailedly analyzed existing transfer learning methods used in DEs problems and propose a feasible POTT method that simultaneously admits generalizability to common DEs and physics preservation of specific problem. Based on the the availability of physical prior, two forms of realization of are introduced to encourage the POTT map to characterize target distribution in a physics-preserved way. A dual optimization problem and the consistency property are formulated to explicitly solve the optimal map. Extensive evaluation and analysis experiments on both simulation and real-world datasets with varying difficulties validate the effectiveness of POTT for data-driven methods in DEs problems.

Limitations and future works. (1) The core of POTT lies in characterizing target distribution with limited samples. However, similar to other distribution-based methods, POTT’s efficacy diminishes when the quantity of available target data is extremely small, rendering its potential in one-shot or few-shot scenarios. Moreover, when abundant target data are accessible, finetuning or even training from scratch is sufficient for cross-domain application. Therefore, POTT is more valuable when the amount of target sample is small, yet not extremely small, as shown in Sec. 5 and Sec. D. (2) The learning of POTT relies on the min-max optimization, which is complex and costly for models with large-scale parameters. The resolution of the experimental data is relatively low, thereby necessitating a less complex model. However, in DEs problems and application scenarios that require high resolution, a larger model is needed to fit POTT. To address this issue, one idea is to leverage the well-developed multimodal large models to reduce the training load of the method, which will further enhance the practicality and broad applicability of this method. We believe this will contribute to the generalizability and universality of the scientific and technological achievements, such as the cross-regional climate forecasting tried in this paper.

Acknowledgments

This work is supported in part by National Key R&D Program of China (2024YFA1011900), National Natural Science Foundation of China (Grant No. 62376291), Guangdong Basic and Applied Basic Research Foundation (2023B1515020004), Science and Technology Program of Guangzhou (2024A04J6413), and the Fundamental Research Funds for the Central Universities, Sun Yat-sen University (24xkjc013).

References

- [1] Arip Asadulaev, Alexander Korotin, Vage Egiazarian, Petr Mokrov, and Evgeny Burnaev. Neural optimal transport with general cost functionals. In *ICLR*, 2024.
- [2] Nikolas Borrel-Jensen, Somdatta Goswami, Allan P. Engsig-Karup, George Em Karniadakis, and Cheol-Ho Jeong. Sound propagation in realistic interactive 3d scenes with parameterized sources using deep neural operators. *Proceedings of the National Academy of Sciences*, 2024.
- [3] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *CVPR*, 2022.
- [4] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 1995.
- [5] Xinyang Chen, Sinan Wang, Jianmin Wang, and Mingsheng Long. Representation subspace distance for domain adaptation regression. In *ICML*, 2021.
- [6] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.
- [7] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013.
- [8] Max Daniels, Tyler Maunu, and Paul Hand. Score-based generative neural networks for large-scale optimal transport. In *NeurIPS*, volume 34, pages 12955–12965, 2021.
- [9] Shaan Desai, Marios Mattheakis, Hayden Joy, Pavlos Protopapas, and Stephen J. Roberts. One-shot transfer learning of physics-informed neural networks. In *ICML AI for Science Workshop*, 2022.
- [10] Jiaojiao Fan, Shu Liu, Shaojun Ma, Hao-Min Zhou, and Yongxin Chen. Neural monge map estimation and its applications. *Transactions on Machine Learning Research*, 2023.
- [11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 2016.
- [12] Somdatta Goswami, Katiana Kontolati, Michael D Shields, and George Em Karniadakis. Deep transfer operator learning for partial differential equations under conditional shift. *Nature Machine Intelligence*, 2022.
- [13] Nikita Gushchin, Alexander Kolesov, Alexander Korotin, Dmitry P Vetrov, and Evgeny Burnaev. Entropic neural optimal transport via diffusion processes. In *NeurIPS*, volume 36, pages 75517–75544, 2023.
- [14] Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *ICML*, 2023.
- [15] Junyan He, Seid Koric, Diab Abueidda, Ali Najafi, and Iwona Jasiuk. Geom-deeponet: A point-cloud-based deep operator network for field predictions on 3d parameterized geometries. *Computer Methods in Applied Mechanics and Engineering*, 2024.

- [16] Pengzhan Jin, Shuai Meng, and Lu Lu. Mionet: Learning multiple-input operators via tensor product. *SIAM Journal on Scientific Computing*, 2022.
- [17] Tanguy Kerdoncuff, Rémi Emonet, Michaël Perrot, and Marc Sebban. Optimal tensor transport. In *AAAI*, 2022.
- [18] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. In *ICLR*, 2023.
- [19] Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations’ operator learning. *Transactions on Machine Learning Research*, 2023.
- [20] Zijie Li, Dule Shu, and Amir Barati Farimani. Scalable transformer for pde surrogate modeling. In *NeurIPS*, 2023.
- [21] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 2023.
- [22] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *ICLR*, 2021.
- [23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.
- [24] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deepnet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 2021.
- [25] Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 2022.
- [26] Yiming Meng, Ruikun Zhou, Amartya Mukherjee, Maxwell Fitzsimmons, Christopher Song, and Jun Liu. Physics-informed neural network policy iteration: Algorithms, convergence, and verification. In *ICML*, 2024.
- [27] Ismail Nejjar, Qin Wang, and Olga Fink. Dare-gram: Unsupervised domain adaptation regression by aligning inverse gram matrices. In *CVPR*, 2023.
- [28] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 2019.
- [29] Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.
- [30] Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large scale optimal transport and mapping estimation. In *ICLR*, 2018.
- [31] Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *NeurIPS*, 2023.
- [32] Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *ICLR*, 2023.
- [33] Yogesh Verma, Markus Heinonen, and Vikas Garg. ClimODE: Climate and weather forecasting with physics-informed neural ODEs. In *ICLR*, 2024.

- [34] Yuchen Wang, Can Xiong, Changjiang Ju, Genke Yang, Yu-wang Chen, and Xiaotian Yu. A deep transfer operator learning method for temperature field reconstruction in a lithium-ion battery pack. *IEEE Transactions on Industrial Informatics*, 2024.
- [35] Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *ICML*, 2024.
- [36] Haixu Wu, Hang Zhou, Mingsheng Long, and Jianmin Wang. Interpretable weather forecasting for worldwide stations with a unified deep model. *Nature Machine Intelligence*, 2023.
- [37] Wuzhe Xu, Yulong Lu, and Li Wang. Transfer learning enhanced deepnet for long-time prediction of evolution equations. In *AAAI*, 2023.
- [38] Hao-Ran Yang, Chuan-Xian Ren, and You-Wei Luo. Cod: Learning conditional invariant representation for domain adaptation regression. In *ECCV*, 2025.
- [39] Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with nowcastnet. *Nature*, 2023.
- [40] Tingtao Zhou, Xuan Wan, Daniel Zhengyu Huang, Zongyi Li, Zhiwei Peng, Anima Anandkumar, John F. Brady, Paul W. Sternberg, and Chiara Daraio. Ai-aided geometric design of anti-infection catheters. *Science Advances*, 2024.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The claims made in the abstract and the end of the introduction accurately summarize the paper's contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations about the method and the experiments, influence factor of the performance, and the future works are discussed in Sec. 6. The discussion on computational efficiency of the algorithm is provided in Appendix B.3.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: The full set of assumptions and proof are provided in Appendix B. And an intuitive understanding of the Thm. 4.3 is provided right after the theorem.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: All details about the data generation and implementation are provided in Sec. 5 and Appendix C. Some experiment settings are based on previous works in different areas, we have made clear instruction in these sections.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: As our work is about a general model transfer method for DEs problems, the reproducibility can be guaranteed by the algorithm description and implementation details provided in Sec. 5, Appendix B.3 and Appendix C.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experimental settings and details are provided in Sec. 5 and Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The std of results are provided.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We indicate the compute resources in Appendix C.3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and have made sure the research conform the Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The potential positive societal impacts are discussed in Sec. 6. As our work is a general model transfer method, the potential negative societal impacts depends on the specific transferred data-driven model and its applications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work is a general model transfer method, the potential risks depends on the specific model that use this method.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The original papers that produced the code package and datasets are properly credited and cited in Sec. 5 and Appendix C. All codes and datasets used for both simulation and real-world datasets takes the MIT License.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No assets are released in this paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Notations

The notations appear in this paper are summarized as follows:

Table 7: Notations.

Notation	Description
$\mathcal{D} = \mathcal{D}_k \times \mathcal{D}_u = \{(k, u)\}$	General function set
$\Omega = \Omega_k \times \Omega_u$	Domain of function
$k = k(x)$	function defined on Ω_k
$u = u(x)$	function defined on Ω_u
$\mathcal{F}(x; k, u)$	Differential Equation
$\mathcal{G} : \mathcal{D}_k \rightarrow \mathcal{D}_u$	Operator map from $\mathcal{D}_k$ to $\mathcal{D}_u$
$P(k, u)$	Product probability function defined on function set $\mathcal{D}$
$p(k, u)$	Product probability density function of function set $\mathcal{D}$
$\mathcal{P}_{\mathcal{D}}$	Set of probability functions defined on $\mathcal{D}$
$\mathcal{D}^s$	Source domain; a subset of $\mathcal{D}$
$\mathcal{D}^t$	Target domain; a subset of $\mathcal{D}$
$\mathcal{G}^s$	Operator relation on $\mathcal{D}^s$
$\mathcal{G}^t$	Operator relation on $\mathcal{D}^t$
P^s	Probability function on $\mathcal{D}^s$
P^t	Probability function on $\mathcal{D}^t$
$T : \mathcal{D}^s \rightarrow \mathcal{D}^t$	Function map between $\mathcal{D}^s$ and $\mathcal{D}^t$
$\mathcal{T}$	Ideal solution of optimal transport problem
$M(P^s, P^t)$	Monge problem between probability P^s and P^t
$K(P^s, P^t)$	Kantorovich problem between probability P^s and P^t
$\mathcal{R}(T)$	Physical regularization on T
$m(\cdot, \cdot)$	Metric defined on $\mathcal{D}_u$
$M_{phy}(P^s, P^t)$	Monge problem with physical regularization
$K_{phy}(P^s, P^t)$	Kantorovich problem with physical regularization
$c(\cdot, \cdot)$	Cost function in OT problem
f	Lagrange multiplier
$\mathcal{L}$	Objective function of optimization problem
(T^*, f^*)	Saddle point solution of dual problem
$\hat{\mathcal{G}}^s$	Approximated operator on $\mathcal{D}^s$
$\hat{\mathcal{G}}^t$	Approximated operator on $\mathcal{D}^t$
$\hat{T}$	Approximation of T

we slightly abuse the notations k and u to represent both functions and their discretized value vectors. The superscript s or t denotes the domain. The subscript k or u denotes the projection of the original product space or distribution.

B Theory and method

B.1 Derivation of dual formula

Given POTT problem

$$\inf_{T \# P^s = P^t} \int_{\mathcal{D}^s} c((k, u), T(k, u)) dP^s + \mathcal{R}(T), \quad (15)$$

where $T = (T_k, T_u)$, $T_k = T|_{\mathcal{D}_k}$, $T_u = T|_{\mathcal{D}_u}$, we reorganize it as a constrained optimization problem:

$$\inf_T \int_{\mathcal{D}^s} c((k, u), T(k, u)) dP^s + \mathcal{R}(T) \quad (16)$$

$$s.t. T \# P^s = P^t \quad (17)$$

Following the dual optimization theory, we introduce the Lagrange multiplier f to construct the Lagrange function:

$$\mathcal{L}(T, f) = \int_{\Omega_k \times \Omega_u} c((k, u), T(k, u)) dP^s + \lambda \mathcal{R}(T) + \int_{\Omega_k \times \Omega_u} f(k, u) d(P^t - T \# P^s). \quad (18)$$

As the physical regularization regularized the pushforward distributions $P^r = T_{\#}P^s$, it can be considered with the source distribution. Then it comes to

$$\mathcal{L}(T, f) = \int_{\Omega_k \times \Omega_u} c((k, u), T(k, u)) - f(T(k, u)) + \lambda \mathcal{R}(T) dP^s + \int_{\Omega_k \times \Omega_u} f(k, u) dP^t. \quad (19)$$

And the dual problem of Eq. (15) is

$$\sup_f \inf_T \mathcal{L}(T, f), \quad (20)$$

which is exactly Eq. (13).

B.2 Proof of Thm. 4.3

We prove Thm. 4.3 based on previous work [10].

Theorem B.1 (Consistency). *Suppose the dual problem Eq. (13) admits at least one saddle point solution, denoted as (T^*, f^*) . Let $\mathcal{L}$ be the objective of Eq. (13). Then*

- the dual problem Eq. (13) equals to the Kantorovich problem with physical regularization in terms of total cost, i.e. $\mathcal{L}(P^s, T^*, f^*) = K(P^s, P^t) + \mathcal{R}(T^*)$.
- if $T_{\#}^* P^s = P^t$, then Eq. (13) degenerates to the dual form of the primal Monge problem Eq. (1), T^* is a Monge map, i.e. $\mathcal{L}(P^s, T^*, f^*) = M(P^s, P^t)$.

Proof. (1) Let $k^r = T_{\theta_k}(k^s)$, $u^r = T_{\theta_u}(u^s)$, the inner optimization problem can be formulated as

$$\begin{aligned} & \inf_T \mathcal{L}(T, f) \\ &= \inf_T \int c((k^s, u^s), T(k^s, u^s)) - f(T(k^s, u^s)) + \lambda \mathcal{R}(T) dP^s + \int f(k^t, u^t) dP^t \\ &= - \int \sup_{(\xi^r, \zeta^r)} \{f(\xi^r, \zeta^r) - [c((k^s, u^s), (\xi^r, \zeta^r)) + \lambda \mathcal{R}(T)]\} dP^s + \int f(k^t, u^t) dP^t \\ &= \int f(k^t, u^t) dP^t - \int f^{c,-}(k^s, u^s) dP^s, \end{aligned} \quad (21)$$

where

$$f^{c,-}(k^s, u^s) = \sup_{(\xi^r, \zeta^r)} (f(\xi^r, \zeta^r) - [c((k^s, u^s), (\xi^r, \zeta^r)) + \lambda \mathcal{R}(T)]) \quad (22)$$

is the c-transform of the physics-regularized Kantorovich dual problem. Then the optimization problem Eq. (13) becomes

$$\sup_f \left[\int f(k^t, u^t) dP^t - \int f^{c,-}(k^s, u^s) dP^s \right], \quad (23)$$

which is exactly the physics-regularized Kantorovich problem.

Therefore, if (T^*, f^*) is the saddle point solution of Eq. (13), then f^* is an optimal solution of Eq. (23), $\mathcal{L}(P^s, T^*, f^*) = K(P^s, P^t) + \mathcal{R}(T^*)$, which verifies the first assertion of the theorem.

(2) The saddle point (T^*, f^*) satisfy

$$T^*(k^s, u^s) \in \argmax_{(\xi^r, \zeta^r)} f^*(\xi^r, \zeta^r) - [c((k^s, u^s), (\xi^r, \zeta^r)) + \lambda \mathcal{R}(T)] \text{ a.s.} \quad (24)$$

$$\implies f^{c,-}(k^s, u^s) = f^*(T^*(k^s, u^s)) - [c((k^s, u^s), T^*(k^s, u^s)) + \lambda \mathcal{R}(T)] \quad (25)$$

where $f^{c,-}(k^s, u^s) = \sup_{(\xi^r, \zeta^r)} (f^*(\xi^r, \zeta^r) - [c((k^s, u^s), (\xi^r, \zeta^r)) + \lambda \mathcal{R}(T)])$.

With condition $T_{\#}^* P^s = P^t$, the pushforward distribution $P^r = P^t$, then from the construction of $\mathcal{R}(T)$, we have $\mathcal{R}(T) = 0$. Thus Eq. (13) degenerates to

$$\sup_f \inf_T \int_{\Omega_k \times \Omega_u} c((k, u), T(k, u)) - f(T(k, u)) dP^s + \int_{\Omega_k \times \Omega_u} f(k, u) dP^t, \quad (26)$$

which is exactly the dual form of the primal Monge problem. Then we have

$$\begin{aligned}
& \int_{\Omega} c((k^s, u^s), T^*(k^s, u^s)) dP^s \\
&= \int_{\Omega} f^*(T^*(k^s, u^s)) dP^s - \int_{\Omega} f^{*, -}(k^s, u^s) dP^s \\
&= \int_{\Omega} f^*(k^t, u^t) dP^t - \int_{\Omega} f^{*, -}(k^s, u^s) dP^s \\
&= \int_{\Omega \times \Omega} f^*(k^t, u^t) - f^{*, -}(k^s, u^s) d\pi \\
&\leq \int_{\Omega \times \Omega} c((k^s, u^s), (k^t, u^t)) d\pi, \quad \forall \pi \in \Pi(P^s, P^t)
\end{aligned} \tag{27}$$

Take infimum on both sides of the inequation, we obtain

$$\begin{aligned}
& \inf_T \int_{\Omega} c((k^s, u^s), T^*(k^s, u^s)) dP^s \\
&\leq \inf_{\pi} \int_{\Omega \times \Omega} c((k^s, u^s), (k^t, u^t)) d\pi \\
&\leq \int_{\Omega} c((k^s, u^s), T(k^s, u^s)) dP^s,
\end{aligned} \tag{28}$$

where T is any map that satisfies $(Id, T)_{\#} P^s = \pi \in \Pi(P^s, P^t)$. Therefore, the solution of the Monge problem exists and T^* is the Monge map. $\square$

B.3 Algorithm analysis

Algorithm 1: Optimization of POTT

- 1 **Input:** source data $\hat{D}_s$, pretrained model $\mathcal{G}_{\eta}$, target data $\hat{D}_t$;
 - 2 Initialize T_{θ}, f_{ϕ} ;
 - 3 **for** N_{11} steps **do**
 - 4 Freeze ϕ , update θ to minimize the first objective of Eq. (14) for N_{12} steps;
 - 5 Freeze θ , update ϕ to maximize the first objective of Eq. (14) with θ ;
 - 6 **end**
 - 7 **for** N_2 steps **do**
 - 8 Freeze θ and ϕ , update η to minimize the second objective in Eq. (14);
 - 9 **end**
 - 10 **Output:** $\mathcal{G}_{\eta}$ as approximation of $\mathcal{G}_t$.
-

Thus, the entire training process of POTT requires $O((N_{11} \cdot N_{12} + N_2)B(C_{\phi} + C_{\eta} + C_{\theta}))$ operations, where $C_{\phi}, C_{\eta}, C_{\theta}$ are model parameters, B is the batchsize, N_{12} is typically set to 10. As the dataset size grows larger, N_{11} and N_2 should be set larger, and the model size of T_{θ} and f_{ϕ} also grow. In summary, the computational efficiency of POTT is similar to other neural OT methods.

C Experiment details

C.1 Simulation datasets

The relative Mean Square Error $rMSE = \|u_{pred} - u_{gt}\|_2^2 / \|u_{gt}\|_2^2$ is reported for evaluation, where u_{gt} denotes the ground truth of output u .

Simulation datasets used for evaluation are generated as follows:

C.1.1 Burgers' equation

Considering the 1-D Burgers' equation on unit torus:

$$u_t + uu_x = \nu u_{xx}, \quad x \in (0, 1), t \in (0, 1], \tag{29}$$

we aim to learn the operator mapping the initial condition to the solution function at time one, i.e. $\mathcal{G}_\theta : u_0 = u(x, 0) \rightarrow u(x, 1)$. As shown in Tab. 8, we differ the generation of u_0 and parameter ν to construct different domains:

Table 8: Generation of u_0 and parameter settings in 1-d Burgers' equation.

SUB-DOMAIN	DESCRIPTION
$\mathcal{D}_1$	$u_0 \sim \mathcal{N}(0, 7^2(-\Delta + 7^2\mathcal{I})^{-2}), \nu = 0.01$
$\mathcal{D}_2$	$u_0 \sim \mathcal{N}(0.2, 49^2(-\Delta + 7^2\mathcal{I})^{-2.5}), \nu = 0.002$
$\mathcal{D}_3$	$u_0 \sim \mathcal{N}(0.5, 625^2(-\Delta + 25^2\mathcal{I})^{-2.5}), \nu = 0.004$

The $\mathcal{N}$ denotes the normal distribution. The resolution of x-axis is 1024.

We assess the domain shift between these domains in a rather straightforward way. Specifically, we regard the data-driven model trained with sufficient target data as **Oracle**, and compare the prediction error between the finetuning method trained with 100 target data and the Oracle. Tasks with lower gap are considered as simpler than tasks with larger gap. As shown in Tab. 9, the prediction gap in tasks $\mathcal{D}_2 \rightarrow \mathcal{D}_1$, $\mathcal{D}_2 \rightarrow \mathcal{D}_3$, $\mathcal{D}_3 \rightarrow \mathcal{D}_1$ are smaller than the other three, which means models are easier to transfer in these tasks. So we only conduct comparison experiment in tasks $\mathcal{D}_1 \rightarrow \mathcal{D}_2$, $\mathcal{D}_1 \rightarrow \mathcal{D}_3$ and $\mathcal{D}_3 \rightarrow \mathcal{D}_2$.

Table 9: Evaluation of tasks' difficulties of Burgers' equation.

METHOD	$\mathcal{D}_1 \rightarrow \mathcal{D}_2$	$\mathcal{D}_1 \rightarrow \mathcal{D}_3$	$\mathcal{D}_2 \rightarrow \mathcal{D}_1$	$\mathcal{D}_2 \rightarrow \mathcal{D}_3$	$\mathcal{D}_3 \rightarrow \mathcal{D}_1$	$\mathcal{D}_3 \rightarrow \mathcal{D}_2$
FINETUNING	0.1191	0.0801	0.0381	0.0786	0.0252	0.0938
ORACLE	0.0402	0.0403	0.0140	0.0403	0.0140	0.0402

As shown in Tab. 2, the improvement of POTT compared to finetuning is substantial. When the amount of target data is only 50, POTT reduced the relative error by 23.64% (from 0.2001 to 0.1528) in task $\mathcal{D}_1 \rightarrow \mathcal{D}_2$ and by 19.21% (from 0.1546 to 0.1249) in task $\mathcal{D}_3 \rightarrow \mathcal{D}_2$. When the amount of target data is 100, although the performance of finetuning greatly improves, POTT still largely reduces the relative error by 18.98% and 19.30% in task $\mathcal{D}_1 \rightarrow \mathcal{D}_2$ and $\mathcal{D}_3 \rightarrow \mathcal{D}_2$. In the relatively simple task $\mathcal{D}_1 \rightarrow \mathcal{D}_3$, although finetuning already achieves satisfactory results, POTT can still reduce the relative error by about 10%, while TL-DeepONet, DARE-GRAM and COD even caused negative transfer and result in larger relative error.

C.1.2 Darcy flow

The 2-d Darcy flow takes the form

$$\begin{aligned} \nabla \cdot (k(x)\nabla u(x)) &= 1, \quad x \in [0, 1] \times [0, 1] \\ u(x) &= 0, \quad x \in \partial([0, 1] \times [0, 1]). \end{aligned} \quad (30)$$

We aim to learn the operator mapping the diffusion coefficient $k(x)$ to the solution function $u(x)$, i.e. $\mathcal{G}_\theta : k(x) \rightarrow u(x)$. We use the leading 100 terms in a truncated *Karhunen – Loève* (KL) expansion for a Gaussian process with zero mean and covariance kernel $\mathcal{K}(x)$ to generate $a(x)$, and construct different function domains by differ the kernel $\mathcal{K}(x, x')$, as shown in Tab. 10.

The Ω_{square} denotes a square with vertex on $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$ in $[0, 1] \times [0, 1]$, $\Omega_{triangle}$ denotes a triangle with vertex on $\{(0, 0), (0, 1), (0.5, 1)\}$ in $[0, 1] \times [0, 1]$. The resolution of $x \in [0, 1] \times [0, 1]$ is 64×64 . Similarly, the difficulties of transfer tasks are shown in Tab. 11. As discussed in Sec. 5, we conduct experiments in the hard tasks $\mathcal{D}_2 \rightarrow \mathcal{D}_1$, $\mathcal{D}_1 \rightarrow \mathcal{D}_3$ and $\mathcal{D}_2 \rightarrow \mathcal{D}_3$. Results are provided in Tab. 3.

C.1.3 Advection equation

The Advection equation takes the form

$$u_t + \nu u_x = 0, \quad x \in (0, 1), t \in (0, 1]. \quad (31)$$

Table 10: Generation of $a(x)$ in Darcy flow.

SUB-DOMAIN	DESCRIPTION
$\mathcal{D}_1$	$\mathcal{K}(x, x') = \exp(-\frac{\ x-x'\ _2^2}{2}), \Omega_u = \Omega_{square}$
$\mathcal{D}_2$	$\mathcal{K}(x, x') = \exp(-\frac{\ x-x'\ _2^2}{2}), \Omega_u = \Omega_{triangle}$
$\mathcal{D}_3$	$\mathcal{K}(x, x') = \exp(-\frac{\ x-x'\ _1^2}{2}), \Omega_u = \Omega_{square}$

Table 11: Evaluation of tasks' difficulties of Darcy flow.

METHOD	$\mathcal{D}_1 \rightarrow \mathcal{D}_2$	$\mathcal{D}_1 \rightarrow \mathcal{D}_3$	$\mathcal{D}_2 \rightarrow \mathcal{D}_1$	$\mathcal{D}_2 \rightarrow \mathcal{D}_3$	$\mathcal{D}_3 \rightarrow \mathcal{D}_1$	$\mathcal{D}_3 \rightarrow \mathcal{D}_2$
FINETUNING	0.0429	0.1605	0.0869	0.3553	0.0469	0.0598
ORACLE	0.0111	0.0682	0.0153	0.0682	0.0153	0.0111

We aim to learn the operator mapping the initial condition to the solution function at a continuous time set $[0, 1]$, i.e. $\mathcal{G}_\theta : u_0 = u(x, 0) \rightarrow u(x, t)$. As shown in Tab. 12, we differ the function types and generation process of u_0 and parameter ν to construct different domains:

Table 12: Generation of u_0 and parameter settings in Advection equation.

SUB-DOMAIN	DESCRIPTION
$\mathcal{D}_1$	$u_0(x) = ax^2 + bx + c, a, b, c \in \mathcal{U}(-1, 1), \nu = 3$
$\mathcal{D}_2$	$u_0(x) = ax^3 + bx^2 + cx + d, a \in \mathcal{U}(0, 1), b, c \in \mathcal{U}(-0.5, 0.5), d = 0.5, \nu = 2$
$\mathcal{D}_3$	$u_0(x) = a \sin(bx + c), a \in \mathcal{U}(0, 1), b \in \mathcal{U}(5, 10), c \in \mathcal{U}(-1, 1), \nu = 1$

The $\mathcal{U}$ denotes the uniform distribution. The resolution of x-axis and t-axis are 100 and 50, respectively. Similarly, the difficulties of transfer tasks are shown in Tab. 13. For this equation, we select three easier tasks to explore the effectiveness of POTT in easy tasks, i.e., the $\mathcal{D}_1 \rightarrow \mathcal{D}_2$, $\mathcal{D}_2 \rightarrow \mathcal{D}_1$ and $\mathcal{D}_3 \rightarrow \mathcal{D}_2$. As shown in Tab. 4, although the performance of finetuning is satisfying, POTT can further reduce the prediction error to a lower level.

C.2 Real-world datasets

We use the preprocessed version of ERA5 from WeatherBench [29] as real-world datasets for climate forecasting task. We utilize the ClimODE [33] as backbone model and conduct experiment based on their settings and codes, which are all available from their paper. Note that we only report the latitude-weighted RMSE for evaluation, since the Anomaly Correlation Coefficient (ACC) reported in [33] is rather closed for most methods in comparison. The definition of the latitude-weighted RMSE is

$$\text{RMSE} = \frac{1}{N} \sum_t \sqrt{\frac{1}{HW} \sum_h \sum_w \alpha(h) (u_{gt} - u_{pred})^2},$$

where $\alpha(h) = \cos(h) / \frac{1}{H} \sum_{h'} \cos(h')$ is the latitude weight, u_{gt} and u_{pred} are ground truth and model prediction respectively. Lower latitude-weighted RMSE values indicate better model performance in capturing spatial or climate patterns. Visualized results and analysis are provided in Sec. 5, details results are provided in Tab. 14.

C.3 Implementation details

Simulation experiment. To test the generalizability of POTT with different models, we employed different backbones on various datasets. On the Burgers' equation dataset, $\mathcal{G}_\eta$ is parametrized as a 1-d Fourier Neural Operator (FNO) model, T_θ is an operator network composed of two fully connected networks (FCN), and f_ϕ is an FCN. On the Advection equation and Darcy flow datasets, $\mathcal{G}_\eta$ adopt

Table 13: Evaluation of tasks’ difficulties of Advection equation.

METHOD	$\mathcal{D}_1 \rightarrow \mathcal{D}_2$	$\mathcal{D}_1 \rightarrow \mathcal{D}_3$	$\mathcal{D}_2 \rightarrow \mathcal{D}_1$	$\mathcal{D}_2 \rightarrow \mathcal{D}_3$	$\mathcal{D}_3 \rightarrow \mathcal{D}_1$	$\mathcal{D}_3 \rightarrow \mathcal{D}_2$
FINETUNING	0.0143	0.0795	0.0891	0.0777	0.2226	0.0723
ORACLE	0.0123	0.0134	0.0252	0.0134	0.0252	0.0153

Table 14: Latitude weighted RMSE($\downarrow$) results of global forecasting on ERA5 dataset. Oracle denotes the model trained with sufficient target data.

	Lead-Time (hours)	ClimODE				ClimaX Oracle
		Source-only	Finetuning	POTT	Oracle	
z	6	265.71 $\pm$ 14.76	216.77 $\pm$ 14.58	196.53 $\pm$ 14.89	102.9 $\pm$ 9.3	247.5
	12	405.83 $\pm$ 27.34	295.89 $\pm$ 22.92	251.4 $\pm$ 24.93	134.8 $\pm$ 162.3	265.3
	18	568.39 $\pm$ 35.93	381.6 $\pm$ 29.34	325.7 $\pm$ 32.28	12.7 $\pm$ 14.4	319.8
	24	723.61 $\pm$ 43.37	445.45 $\pm$ 35.35	374.68 $\pm$ 39.66	193.4 $\pm$ 16.3	364.9
	36	1127.46 $\pm$ 58.67	598.61 $\pm$ 44.72	506.43 $\pm$ 51.64	259.6 $\pm$ 22.3	455.0
t	6	1.81 $\pm$ 0.07	1.63 $\pm$ 0.07	1.36 $\pm$ 0.07	1.16 $\pm$ 0.06	1.64
	12	2.72 $\pm$ 0.14	2.17 $\pm$ 0.09	1.85 $\pm$ 0.09	1.32 $\pm$ 0.13	1.77
	18	3.51 $\pm$ 0.19	2.5 $\pm$ 0.1	1.89 $\pm$ 0.11	1.47 $\pm$ 0.16	1.93
	24	4.51 $\pm$ 0.26	2.71 $\pm$ 0.11	1.91 $\pm$ 0.12	1.55 $\pm$ 0.18	2.17
	36	6.96 $\pm$ 0.42	3.08 $\pm$ 0.14	2.54 $\pm$ 0.19	1.75 $\pm$ 0.26	2.49
t2m	6	9.09 $\pm$ 0.21	3.43 $\pm$ 0.24	2.65 $\pm$ 0.2	1.21 $\pm$ 0.09	2.02
	12	9.83 $\pm$ 0.24	4.09 $\pm$ 0.25	3.1 $\pm$ 0.21	1.45 $\pm$ 0.10	2.26
	18	9.97 $\pm$ 0.21	3.66 $\pm$ 0.14	2.89 $\pm$ 0.2	1.43 $\pm$ 0.09	2.45
	24	9.69 $\pm$ 0.36	3.01 $\pm$ 0.17	2.01 $\pm$ 0.31	1.40 $\pm$ 0.09	2.37
	36	10.91 $\pm$ 0.67	3.86 $\pm$ 0.27	3.13 $\pm$ 0.28	1.70 $\pm$ 0.15	2.87
u10	6	2.22 $\pm$ 0.11	2.19 $\pm$ 0.11	2.01 $\pm$ 0.11	1.41 $\pm$ 0.07	1.58
	12	3.09 $\pm$ 0.15	2.71 $\pm$ 0.12	2.42 $\pm$ 0.13	1.81 $\pm$ 0.09	1.96
	18	3.8 $\pm$ 0.16	3.11 $\pm$ 0.13	2.6 $\pm$ 0.13	1.97 $\pm$ 0.11	2.24
	24	4.48 $\pm$ 0.17	3.36 $\pm$ 0.13	2.75 $\pm$ 0.14	2.01 $\pm$ 0.10	2.49
	36	6 $\pm$ 0.21	4.05 $\pm$ 0.15	3.24 $\pm$ 0.17	2.25 $\pm$ 0.18	2.98
v10	6	2.45 $\pm$ 0.15	2.34 $\pm$ 0.13	2.15 $\pm$ 0.14	1.53 $\pm$ 0.08	1.60
	12	3.4 $\pm$ 0.22	2.88 $\pm$ 0.14	2.6 $\pm$ 0.16	1.81 $\pm$ 0.12	1.97
	18	4.03 $\pm$ 0.26	3.15 $\pm$ 0.14	2.73 $\pm$ 0.17	1.96 $\pm$ 0.16	2.26
	24	4.58 $\pm$ 0.25	3.44 $\pm$ 0.15	2.86 $\pm$ 0.19	2.04 $\pm$ 0.10	2.48
	36	5.72 $\pm$ 0.25	4.21 $\pm$ 0.18	3.36 $\pm$ 0.22	2.29 $\pm$ 0.24	2.98

a 2-d DeepONet model, T_θ has a structure similar to $\mathcal{G}_\eta$, and f_ϕ is a convolutional neural network (CNN). We use Adam as optimizer and the learning rate is $1e-3$ for all tasks. The learning rate of the backbone of $\mathcal{G}_\eta$ is ten times smaller than the last two layers, which is a widely-used technique in transfer learning. A cosine annealing strategy is adopted for learning rate of $\mathcal{G}_\eta$. Details of model architectures and data generation can be found in codes provided by [22, 25].

Climate forecasting. The architecture and the training details of the data-driven model $\mathcal{G}_\eta$ are consistent with the codes provided by [33]. The architecture and training procedures of T_θ and f_ϕ are the same with that in the simulation experiment.

All experiments are conducted on a single 16GB NVIDIA 4080 device.

D Further discussion

D.1 Practical value of POTT

As the absolute errors of POTT are still too large for practical applications, we discuss the practical value of POTT compared to finetuning with more target data.

- POTT is proposed as a model-agnostic transfer learning framework designed to improve performance under limited supervision. As is standard in transfer learning literature, our primary focus is on the relative gain over baselines such as source-only and finetuning, rather

$\mathcal{D}_1 \rightarrow \mathcal{D}_3$	200	400	600	800	1000
FT	0.1482	0.1391	0.1001	0.0928	0.0867
POTT	0.1309	0.1195	0.0899	0.0817	0.0792

Table 15: Results in the extreme few-shot settings.

$\mathcal{D}_2 \rightarrow \mathcal{D}_3$	200	400	600	800	1000
FT	0.2339	0.1668	0.1267	0.1055	0.0947
POTT	0.2086	0.1487	0.1007	0.0914	0.0834

Table 16: Results in the extreme few-shot settings.

than the absolute error, which also depends on the model’s expressiveness. In our experiments, all methods use the same architecture to ensure a fair comparison.

- Similar evaluation protocols are widely adopted in transfer learning research. For example, in domain adaptation tasks on DomainNet using ResNet-101, SOTA methods achieve only about 40% absolute accuracy. Nevertheless, they are considered effective due to their consistent improvement over source-only baselines (about 25%). This demonstrates that the relative improvement is a well-accepted metric for evaluating transfer methods.
- It seems that the errors of about 0.15 and 0.21 in Darcy flow with 50 and 100 target samples are still too large for practical applications. However, this is mainly due to the limited capacity of the baseline model. Even with 2000 labeled samples in domain $\mathcal{D}_3$, the fully supervised model reaches an error of about 0.07. This highlights the inherent difficulty of the task. To investigate further, we conducted scaling experiments as shown in Tab. 15 and Tab. 16. The results confirm that while increasing the amount of labeled data does improve performance, POTT can achieve comparable accuracy with significantly fewer samples. For instance, POTT with 600 800 samples matches finetuning with 1000 samples, which is very close to the oracle performance achieved by the backbone model (0.07).
- In many practical applications such as high-resolution simulations, chaotic weather systems, or medical diagnostics, collecting a moderate number of labeled samples is feasible, but obtaining large-scale annotations remains prohibitively expensive. In such scenarios, finetuning alone may fall short, and incorporating principled transfer methods like POTT can effectively compensate for the data scarcity.

In summary, while achieving very low absolute error (e.g., <0.05) may require more powerful models or larger datasets, our results clearly show that POTT significantly improves performance in the low-data regime, reducing the required number of labeled target samples to achieve similar accuracy as finetuning. This highlights its practical value in scenarios where labeled data is limited but not extremely scarce — a common situation in many scientific and engineering domains.

D.2 Relations between the reported evaluation metric and the residuals.

While the evaluation metric used in our experiments is the relative L_2 error, it is closely related to the residuals of the governing equations, which is widely used in PDEs learning literature. In fact, it is equivalent to evaluating the residuals of the equations.

For the simulation datasets (e.g., Burgers, Darcy), the ground-truth solutions are generated by high-accuracy numerical solvers, and thus satisfy the PDEs with negligible error. For the advection equation, the ground truth is given by its analytical solution. Given this, the error $du = u_{gt} - u_{pred}$ reflects the deviation from the true solution. By the differential mean value theorem, we have

$$F(u_{pred}) = F(u_{gt}) + F'(u^*)(u_{pred} - u_{gt}) = F'(u^*)du, \quad (32)$$

for some u^* between u_{gt} and u_{pred} . Here, F is the differential operator defined by the PDE. This shows that the residual $\bar{F}(u_{pred})$ is proportional to the error du , with the proportionality factor being the derivative of the operator at some intermediate point. Therefore, our reported prediction errors provide an equivalent proxy for assessing whether the transferred model respects the underlying physics.

D.3 Training time of POTT

The main computational overhead of POTT lies in training the transport map T , but this cost is both amortizable and reducible.

Amortization: The cost of training T is relatively independent of the cost of training the downstream model $\mathcal{G}$. In scenarios where $\mathcal{G}$ is large or expensive to train—as is often the case with real-world scientific models—the relative overhead of POTT becomes modest. We report approximate wall-clock training times on Darcy flow and ERA5 datasets in Tab. 17. As shown, POTT introduces a moderate additional cost, which is acceptable given the substantial improvements in generalization. On simpler tasks such as Darcy, where training $\mathcal{G}$ is very fast, the overhead is more visible, but still within practical limits.

Darcy flow ERA5	Finetuning 5min 80min	COD 7min -	DAREGRAM 11min -	POTT 20min 100min
--------------------	-----------------------------	------------------	------------------------	-------------------------

Table 17: Wall-clock training time comparison.

As shown, POTT introduces a moderate additional cost, which is acceptable given the substantial improvements in generalization. On simpler tasks such as Darcy, where training $\mathcal{G}$ is very fast, the overhead is more visible, but still within practical limits.

Reducibility: As noted in the limitations section, a promising direction for reducing the training cost of POTT is to leverage large pre-trained models from related domains. For example, incorporating pre-trained multimodal foundation models, and applying parameter-efficient tuning or distillation techniques to train the POTT map, can significantly reduce the overhead while potentially improving performance.

D.4 How sensitive are results to the choice of metric and to the accuracy of the surrogate operator ?

Choice of metric m : We use the metric m in Eq. (12) to quantify the distributional discrepancy between P_u^r and $\hat{\mathcal{G}}(P_k^r)$. In principle, m can be any nonparametric divergence or distance measure, such as the Wasserstein distance. In our implementation, we adopt the L2 metric for its simplicity and computational efficiency. Empirically, we observed that L2 provides stable gradients and effective regularization. Preliminary experiments with alternative choices (e.g., L1) yield comparable trends, suggesting robustness to the choice of metric. Nevertheless, systematically studying the impact of different metrics is an important direction for future work.

Accuracy of surrogate operator $\hat{\mathcal{G}}$: Following standard transfer learning assumptions, we rely on a well-trained source model $\hat{\mathcal{G}}$ that captures the physics in the source domain. If $\hat{\mathcal{G}}$ is poorly trained, it may introduce misleading inductive biases. To mitigate this, we ensure that $\hat{\mathcal{G}}$ is trained with sufficient labeled data and validated carefully before transfer. In practice, we also monitor the influence of the marginal-consistency term during training and observe that its contribution is stable, suggesting that the learned physics from the source is indeed beneficial under the domain shift.